ESearch: Incorporating Text Corpus and Structured Knowledge for Open Domain Entity Search

Denghao Ma
School of Information
Renmin University of China
madenghao@ruc.edu.cn

Yueguo Chen*
School of Information
Renmin University of China
chenyueguo@ruc.edu.cn

Jun Chen
School of Information
Renmin University of China
chenjun2013@ruc.edu.cn

Xiaoyong Du
DEKE Key lab (MOE)
Renmin University of China
duyong@ruc.edu.cn

Xiangliang Zhang
KAUST
Thuwal, Saudi Arabia, 23955
xiangliang.zhang@kaust.edu.sa

ABSTRACT

The paper introduces an open domain entity search system called ESearch, which aims at finding a list of relevant entities to an open domain entity search query (a natural language question). The system is built on top of a Wikipedia text corpus, as well as the structured DBPedia knowledge base. Entities are initially ranked by a model which effectively associates context matching (based on the contexts of entities in the unstructured text corpus) and category matching (based on the types of entities in the structured knowledge base). They are ranked further by a re-ranking component supported by blind feedback or user feedback on entities. We show that category matching is critical for the search performance and the re-ranking component can boost the performance largely. Category matching therefore needs some query entity types (especially specific entity types) as input. However, it is often hard for systems to detect specific entity types because users may not be familiar with how the types of desired entities are defined in the structured knowledge base. In ESearch, we design an effective ranking model of entity types to facilitate blind feedback and user feedback on desired entity types for category matching, so that users can effectively perform entity search without the need of explicitly providing any query entity types as inputs.

Keywords

Information retrieval, Entity search, Type ranking

1. INTRODUCTION

Entity search is to retrieve a ranked list of named entities of target types to a given query [3]. This is a large difference from existing general search engines whose target is to retrieve a list of relevant documents. For instance, for an entity search query *works by Charles Rennie Mackintosh*, the desired results may include *Glasgow School of Art*, *Queen's Cross Church*, *Willow Tearooms*, etc., which are all buildings and structures designed by Charles Rennie Mackintosh. Entity search has a wide range of applications such as question answering systems and knowledge services [11].

There have been a stream of solutions of entity search. Some early solutions mainly take a voting strategy [5, 9]. Those models [5, 9] rank entities simply based on their contexts in the relevant documents retrieved from a text corpus, where the document relevance can be treated as a global vote and the document-candidate association can be treated as a local weight. These context matching solutions simply use unstructured text corpus. They are inadequate for entity search because of the ignorance of entity types, which can be easily exploited from many structured knowledge bases such as DBpedia [2]. Recently, the importance of category matching (matches of entity types to the query) has been verified by some solutions of entity search [1, 6].

In the work [3], we proposed a solution which applies and extends the existing context matching model and improves the search performance by combining context matching and category matching more effectively using language models. In this paper, we further propose a type ranking method to recommend some specific and relevant entity types as input of category matching, so that users do not have to input query entity types. This demo is built on top of an unstructured text corpus and a structured knowledge base to demonstrate 1) the importance of category matching on entity search; 2) the effectiveness of the type ranking; 3) the importance of re-ranking component to improve the entity search performance.

The first advantage of ESearch is that, as far as we know, it achieves the state-of-the-art performance on entity search. In general, the basic context matching is applied firstly to retrieve relevant documents from the text corpus, based on the long-range context matching model introduced in [3]. ESearch will exploit query entity types to boost the search performance, based on the category matching in [3]. The ranking model of ESearch effectively incorporates the search results from a text corpus (for context matching) and those from a structured knowledge base (for category matching).

*Yueguo Chen is the corresponding author.

WWW 2017 Companion, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
DOI:http://dx.doi.org/10.1145/3041021.3054720

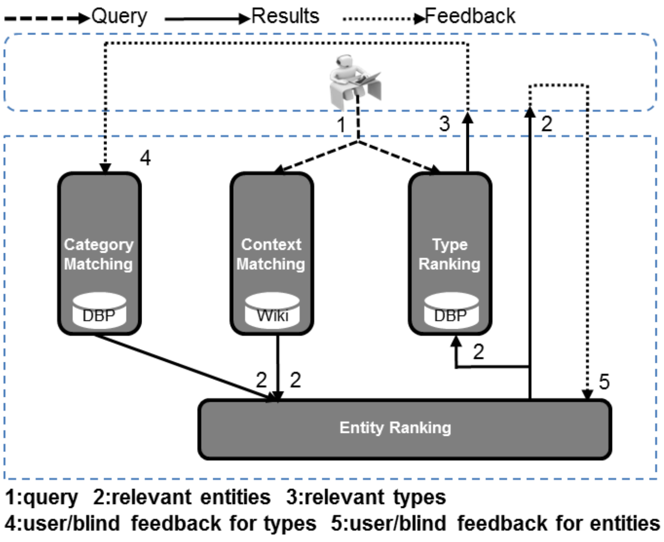

Query
Results
Feedback
1
3
2
4
Category Matching
DBP
Context Matching
Wiki
Type Ranking
DBP
2
2 2
5
Entity Ranking
1:query 2:relevant entities 3:relevant types
4:user/blind feedback for types 5:user/blind feedback for entities

Figure 1: System architecture

It further re-ranks the fused results based on a blind feedback mechanism. Furthermore, if users provide feedback on some resulting entities, ESearch further benefits from a result re-ranking mechanism.

Another advantage of ESearch is that it introduces an effective ranking model of entity types, which provides more specific and relevant entity types for blind or user feedback for category matching. According to the ranking model proposed in [3], in general, the more specific types that can be given by users, the higher the precision that is likely to be achieved. However, it is often hard for users to provide specific and effective entity types because they may not be familiar with how entity types are defined in the structured knowledge base. To address it, we propose an effective ranking method to recommend entity types.

2. SYSTEM IMPLEMENTATION

As depicted in Figure 1, the ESearch system consists of 4 major components: *context matching*, *category matching*, *entity ranking* and *type ranking*. Upon receiving a query, the *context matching* component firstly retrieves top relevant entities from the unstructured text corpus. The *category matching* component generates the relevance of entities based on their category matches to the query entity types (provided either by blind feedback or user feedback). The *entity ranking* component effectively associates the entity relevance from *context matching* and *category matching* by applying techniques proposed in [3]. A re-ranking technique [3] is also supported by *entity ranking*, so that blind or user entity feedback can also help to boost the search performance. The *type ranking* component is to provide a list of relevant entity types/categories for user feedback or blind feedback (applied only when the user does not provide any feedback on entity types), based on the query and the relevant entities generated from the *entity ranking* component.

2.1 Context Matching

This component is built on top of a large corpus of unstructured texts from a Wikipedia dump. Firstly, we use an open source toolkit called Wikipedia-Miner [8] to recognise and extract entity mentions from all the Wiki pages. As a result, all extracted mentions from a document are recorded and indexed. Then, inverted indexes are created for the Wiki corpus to support the efficient online retrieval of relevant documents. A standard language model is applied to rank and retrieve top relevant Wiki documents. Entities embedded in the retrieved documents are then retrieved and ranked based on the voting score from the documents and their local weight in documents.

2.2 Category Matching

The *category matching* component evaluates the relevance of entities based on the matches of entity types to the query entity types, which are provided by either user feedback or blind feedback (where top-k relevant entity types computed from *type ranking* are used as inputs). By using the DBpedia knowledge base, the types of an entity can be extracted from the *type* predicate and the *subject* predicate of the entity. According to our previous study [3], category matching is achieved by extracting the head word (specifying a general category) and some qualifiers (modifying the general category so that the entity type can be more specific) from the entity type and the query type. The entity type can be matched by the query type only they have the same head word and one type contains all the qualifiers of the other. Finally, entities whose type matches to a specified query type are likely to obtain a high relevant score. The relevant entities are then fed to the *entity ranking* component for further ranking using a more comprehensive model.

2.3 Entity Ranking

According to the entity model proposed in [3], the *entity ranking* component associates the relevance of entities generated from *context matching* and *category matching* by simply multiplying them. In addition, as proposed in the entity model [3], a re-ranking strategy is quite important for boosting the performance of entity search. The basic idea of entity re-ranking is to achieve the coherence of entity types among the top results by using top resulting entities as blind feedback to re-rank the search results. Consequently, the relevance of entities having more common types to the feedback entities will be boosted. ESearch applies a small number (20 by default) of top resulting entities for blind feedback. Moreover, ESearch also allows users to provide explicit feedback on whether some search results are relevant entities or not. Those positive feedback entities will be assigned with larger weights in the re-ranking model. On the other hand, those negative feedbacks will be removed from the list of blind feedbacks.

2.4 Type Ranking

A key novelty of this work is on *type ranking*, which takes the query (keywords) and the resulting entities as input and generates a ranked list of relevant entity types so that some relevant entity types can be used by blind feedback or user feedback. This is supported by a ranking model of entity types. Given the query q (contains multiple keywords), the relevance of the entity type t is evaluated as the product of two parts:

$$r(t, q) = r_1(E(t), E(q)) \times r_2(H(t), H(q))$$

where $E(t)$ is a set of entities which belong to the type t and $E(q)$ is a set of entities generated from the *entity ranking* component. $H(\cdot)$ is a function which detects headword from a short text, based on the method of [10]. $r_1(E(t), E(q))$ is to evaluate the relevance from an entity point of view, while $r_2(H(t), H(q))$ is to evaluate the relevance from a headword point of view, which is based on the hypernym-hyponym relationship between the type headword $H(t)$ and the query headword $H(q)$. The component $r_1(E(t), E(q))$ is further evaluated as:

$$r_1(E(t), E(q)) = \frac{|E(t) \bigcap E(q)|}{|E(t)|}$$

For each pair of Wikipedia categories that have hypernym-hyponym relationship, we extract their headwords and build a pair of headwords that also have the hypernym-hyponym relationship. We use $< h_1,\ h_2 >$ to indicate that the headword h_1 is the hypernym of the headword h_2. Statistically, we can compute the relevance between the headwords as follows:

$$r_2(H(t), H(q)) = \frac{n(< H(q), H(t) >)}{\sum_{t' \in T} n(< H(t'), H(t) >)}$$

where $n(< H(q), H(t) >)$ is frequency of the pair $<H(q), H(t)>$ in the Wikipedia category system. Note that $< h, h >$ always holds, and T stands for the set of all Wikipedia entity categories.

2.5 Experimental Evaluation

The target of our experiments is threefold: 1) to test the effectiveness of the *type ranking* component for recommending the specific entity types; 2) to testify the importance of category matching to the performance of entity search; 3) to verify that reranking component can largely improve the performance. The experiment is based on the INEX 2009 entity ranking task (shorted as INEX-ER) [4] that contains 55 entity search queries. The experimental results are shown in Table 1 where C only applies context matching for entity search and other solutions apply both context matching and category matching. B_k adopts the blind feedback strategy, taking top-k relevant types from *type ranking* component as the inputs of *category matching*. H_n adopts user feedback strategy, where three users are involved to pick desired types from top-n results of *type ranking* component as the inputs of *category matching*. R denotes that the re-ranking strategy (based on a blind feedback of entities) is applied in *entity ranking*.

Comparing the results between C and B_k, we see that using blind feedback of entity types can largely improve the performance, even though the users do not provide any query entity types. Reasonably, the best performance of blind feedback is achieved by taking the top-1 type as blind feedback. Such a performance has been better than the reported results of the other work [6] where explicit query entity types are used as inputs. For results of H_n, we say that the user feedback can further improve the search performance by using more effective entity types as the input of *category matching*. In general, it shows that the type ranking model recommends entity types very effectively. By further comparing the results of B_1R with B_1, and $H_{10}R$ with H_{10}, we see the effectiveness of the re-ranking component.

In comparison, we implement two other solutions of entity type ranking. According to the results, the perforamnce of the DGQ model in [7] (Base1) is far from the best performance achieved by our type ranking model because DGQ ignores the hypernym-hyponym relationship between the type and the query headwords. According to the method [10], we detect the headword from a query as the query entity type and annotate it as *Base*2. The search performance is also far from that of our entity ranking model because 1) the query headword may be not an effective entity type since many relevant entities do not belong to it; 2) the query headword is so general that it cannot effectively constrain the types of relevant entities.

Table 1: The effectiveness of type ranking

Solution	MRR	p@5	p@10	p@20	R-pre	xinfAP
C	.185	.080	.098	.118	.120	.085
B_1	.662	.473	.449	.377	.348	.329
B_3	.592	.455	.424	.376	.358	.315
B_5	.577	.418	.405	.361	.341	.305
B_{10}	.499	.327	.318	.289	.280	.244
B_1R	.676	.495	.460	.391	.374	.356
H_3	.633	.491	.467	.387	.375	.342
H_5	.708	.520	.485	.405	.381	.358
H_{10}	.704	.531	.491	.408	.389	.362
$H_{10}R$	.712	.549	.504	.416	.403	.392
*Base*1 [7]	.224	.120	.144	.146	.147	.138
*Base*2 [10]	.430	.265	.271	.244	.226	.197

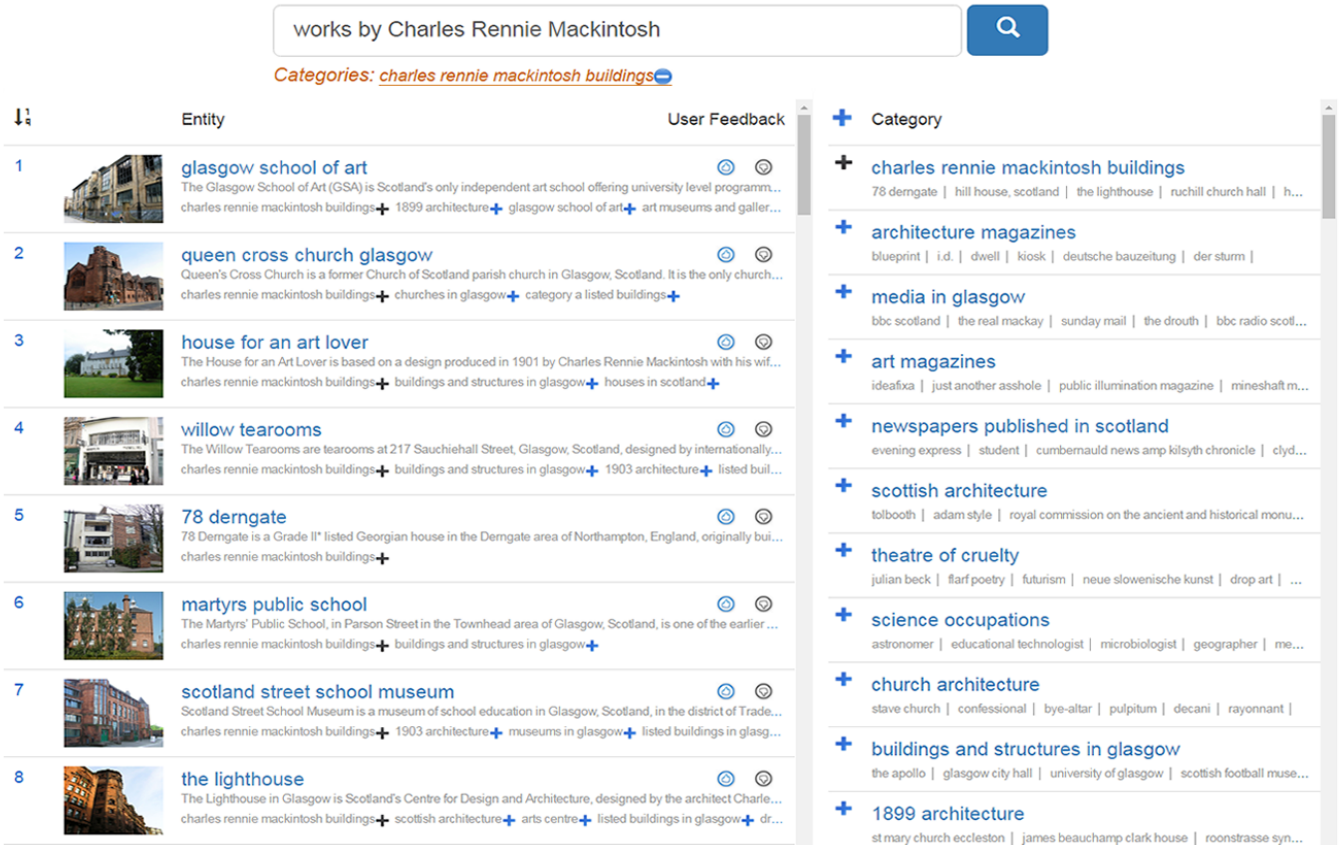
3. DEMONSTRATION

This demonstration generates a list of relevant entities to a given user query. It offers users some friendly interfaces to explore the abstract and the categories of entities. Figure 2 shows the interface of ESearch, which consists of three parts: the query panel (on the top), the category panel (on the right), and the entity panel (on the left). Users initiate a search session by inputting some keywords (or a natural language question) in the input box of the query panel. After that, the category panel will present a list of recommended categories, and the entity panel will show a list of relevant entities (with blind feedback). To facilitate the understanding of search results, on the category panel, we show some representative entities for each recommended category. On the entity panel, we show some typical types and a brief introduction of each resulting entity.

Users then can provide feedbacks on the resulting categories or entities. An entity type feedback can be achieved by clicking a plus icon on the category panel or on the entity panel. The selected entity types will be automatically listed on the query panel, and they can also be removed by clicking the minus icon. Users can also provide feedbacks on the resulting entities by clicking the corresponding icons (either for positive feedbacks or for negative feedbacks). Besides, the blind feedback on entities or types will be adopted, if users click on the blind feedback icons of entity panel or category panel.

During the demonstration, we allow the audience to input any query for finding a list of relevant entities. However, the audience will be suggested to issue queries whose answers are likely to be included in the Wikipedia corpus. Considering that the original search results (without user feedback for types and entities) may not be good enough, we will then suggest the audience to provide some feedback on the entity types or on the resulting entities.

To verify the effectiveness of ESearch, we will provide all the test cases used in the experiments of Table 1. For example, the audience may use a question *works by Charles Ren-*



works by Charles Rennie Mackintosh

Categories: charles rennie mackintosh buildings

↓		Entity	User Feedback
1		glasgow school of art The Glasgow School of Art (GSA) is Scotland's only independent art school offering university level programm... charles rennie mackintosh buildings+ 1899 architecture+ glasgow school of art+ art museums and galler...	
2		queen cross church glasgow Queen's Cross Church is a former Church of Scotland parish church in Glasgow, Scotland. It is the only church... charles rennie mackintosh buildings+ churches in glasgow+ category a listed buildings+	
3		house for an art lover The House for an Art Lover is based on a design produced in 1901 by Charles Rennie Mackintosh with his wif... charles rennie mackintosh buildings+ buildings and structures in glasgow+ houses in scotland+	
4		willow tearooms The Willow Tearooms are tearooms at 217 Sauchiehall Street, Glasgow, Scotland, designed by internationally... charles rennie mackintosh buildings+ buildings and structures in glasgow+ 1903 architecture+ listed buil...	
5		78 derngate 78 Derngate is a Grade II* listed Georgian house in the Derngate area of Northampton, England, originally bui... charles rennie mackintosh buildings+	
6		martyrs public school The Martyrs' Public School, in Parson Street in the Townhead area of Glasgow, Scotland, is one of the earlier ... charles rennie mackintosh buildings+ buildings and structures in glasgow+	
7		scotland street school museum Scotland Street School Museum is a museum of school education in Glasgow, Scotland, in the district of Trade... charles rennie mackintosh buildings+ 1903 architecture+ museums in glasgow+ listed buildings in glasg...	
8		the lighthouse The Lighthouse in Glasgow is Scotland's Centre for Design and Architecture, designed by the architect Charle... charles rennie mackintosh buildings+ scottish architecture+ arts centre+ listed buildings in glasgow+ dr...	

+	Category
+	charles rennie mackintosh buildings 78 derngate | hill house, scotland | the lighthouse | ruchill church hall | h...
+	architecture magazines blueprint | i.d. | dwell | kiosk | deutsche bauzeitung | der sturm |
+	media in glasgow bbc scotland | the real mackay | sunday mail | the drouth | bbc radio scotl...
+	art magazines ideafixa | just another asshole | public illumination magazine | mineshaft m...
+	newspapers published in scotland evening express | student | cumbernauld news amp kilsyth chronicle | clyd...
+	scottish architecture tolbooth | adam style | royal commission on the ancient and historical monu...
+	theatre of cruelty julian beck | flarf poetry | futurism | neue slowenische kunst | drop art | ...
+	science occupations astronomer | educational technologist | microbiologist | geographer | me...
+	church architecture stave church | confessional | bye-altar | pulpitum | decani | rayonnant |
+	buildings and structures in glasgow the apollo | glasgow city hall | university of glasgow | scottish football muse...
+	1899 architecture st mary church eccleston | james beauchamp clark house | roonstrasse syn...

Figure 2: Search interface of ESearch

nie Mackintosh to retrieve buildings or structures designed by Charles Rennie Mackintosh. The system automatically identifies the requirements, and recommends the audience some relevant categories such as *charles rennie mackintosh buildings* and *buildings and structures in glasgow*. The audience can then select some recommended entity types for type feedback. If the audience does not satisfy with the results, he can select some desired entities as positive feedbacks or some others as negative feedbacks to re-rank the results. We also implement the baseline solutions studied in the experiments to allow users to compare the performance of these solutions.

4. CONCLUSION

Category matching is important for the performance of entity search. An effective solution of entity search can be created by associating context matching (based on an unstructured text corpus) and category matching (based on a structured knowledge base). However, finding effective entity types as the inputs of category matching is a challenging task. ESearch effectively addresses this challenge by using a ranking model for entity types with the query and the resulting entities as inputs of type ranking. It shows that both blind feedback and a simple user feedback of entities or their types can largely improve the search performance.

5. ACKNOWLEDGEMENT

This work is supported by the National Science Foundation of China under grant (No. 61472426 and 61432006), 863 key project under grant No. 2015AA015307, the open research program of State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Science (No. CARCH201510), and the ECNU-RUC-InfoSys Joint Data Science Lab.

6. REFERENCES

[1] K. Balog, M. Bron, and M. de Rijke. Category-based query modeling for entity search. In *ECIR*, pages 319–331, 2010.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - A crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165, 2009.

[3] Y. Chen, L. Gao, S. Shi, X. Du, and J. Wen. Improving context and category matching for entity search. In *AAAI*, pages 16–22, 2014.

[4] G. Demartini, T. Iofciu, and A. P. de Vries. Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation, 8th International Workshop of INEX*, pages 254–264, 2009.

[5] Y. Fang, L. Si, Z. Yu, Y. Xian, and Y. Xu. Entity retrieval with hierarchical relevance model. In *TREC*, 2009.

[6] R. Kaptein and J. Kamps. Exploiting the category structure of wikipedia for entity ranking. *Artif. Intell.*, 194, 2013.

[7] S. Liang and M. de Rijke. Formal language models for finding groups of experts. *Inf. Process. Manage.*, 52(4):529–549, 2016.

[8] D. N. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, pages 509–518, 2008.

[9] R. L. T. Santos, C. Macdonald, and I. Ounis. Voting for related entities. In *RIAO*, pages 1–8, 2010.

[10] Z. Wang, H. Wang, and Z. Hu. Head, modifier, and constraint detection in short texts. In *ICDE*, pages 280–291, 2014.

[11] G. Weikum. Search for knowledge. In *SeCO Workshop*, pages 24–39, 2009.